

IAP20 Rec'd PCT/TO 31 JAN 2006

Beschreibung

Verfahren zur Diagnose von Treiberausgängen und Diagnosepulsmanager

5

Die Erfindung betrifft ein Verfahren zur Diagnose von Treiberausgängen, insbesondere zum Einsatz in einem Kraftfahrzeug, bei dem zum Auslesen eines Diagnoseergebnisses an einem Treiberausgang dem Treiberausgang ein Puls zugeführt wird.

10

Die Erfindung betrifft weiterhin einen Diagnosepulsmanager zum Zuführen von Pulsen zu Treiberausgängen in Abhängigkeit von Systemanforderungen, insbesondere zum Einsatz in einem Kraftfahrzeug, um an Treiberausgängen Diagnoseergebnisse aus-

15

zulesen.

Es ist bekannt, Endstufentreiberausgänge im Rahmen einer Fehlerdiagnose beim Betrieb eines Kraftfahrzeugs zu überwachen. Um in gewissen Schaltzuständen eindeutige Diagnoseergebnisse zu erhalten, ist es notwendig, einem Endstufentreiberausgang

20

aktiv einen Puls zuzuführen. Die Zuführung eines solchen Pulses wird vom System angefordert.

25

In diesem Zusammenhang ist es problematisch, wenn für mehrere Treiberausgänge gleichzeitig die Anforderung eines Pulses vorliegt, insbesondere dann, wenn aus Systemgründen nicht gleichzeitig für alle Treiberausgänge ein Puls zur Verfügung gestellt werden kann. Dies ist beispielsweise dann der Fall, wenn das System nur eine Timerfunktionalität zur Verfügung

30

stellt.

Der Erfindung liegt die Aufgabe zugrunde, ein Verfahren und einen Diagnosepulsmanager zur Verfügung zu stellen, so dass bei gleichzeitiger Anforderung von Pulsen für mehrere Treiberausgänge eine zuverlässige Fehlerdiagnose erfolgen kann.

35

Diese Aufgabe wird mit den Merkmalen der unabhängigen Ansprüche gelöst.

Vorteilhafte Ausführungsformen der Erfindung sind in den abhängigen Ansprüchen angegeben.

Die Erfindung baut auf dem gattungsgemäßen Verfahren dadurch auf, dass eine für einen Treiberausgang bestehende Anforderung eines Pulses gespeichert werden kann, wobei insbesondere
10 mehrere gleichzeitig bestehende Anforderungen gespeichert werden können, und dass mehrere gespeicherte Anforderungen gemäß vorgegebener Regeln nacheinander berücksichtigt werden. Liegen somit mehrere Anforderungen für Pulse vor, so ist es
15 nicht erforderlich, nur eine dieser Anforderungen willkürlich zu berücksichtigen und die anderen Anforderungen zu verwerfen. Vielmehr können die Anforderungen gespeichert werden und nachfolgend gemäß vorgegebener Regeln nacheinander berücksichtigt werden.

20 Das Verfahren ist in vorteilhafter Weise dadurch weitergebildet, dass die Anforderungen in einem Diagnosepulsanforderungsregister als binäre Werte gespeichert werden können, dass die in dem Diagnosepulsanforderungsregister gespeicherten Anforderungen in ein Diagnosepulsausführungsregister über-
25 tragen werden können und dass das Diagnosepulsanforderungsregister nach dem Übertragen der Anforderungen in das Diagnosepulsausführungsregister gelöscht wird. Die Systemanforderungen können somit zunächst in einem Diagnosepulsanforderungsregister gespeichert werden und dann in Abhängigkeit
30 von Vorgängen in der Vergangenheit beziehungsweise von sonstigen Kriterien gegebenenfalls in ein Diagnosepulsausführungsregister überführt werden. Nach Überführung der Information kann das Diagnosepulsanforderungsregister gelöscht werden, um so in einem nächsten Zyklus neue Systemanforderungen
35 berücksichtigen zu können.

- Nützlicherweise ist vorgesehen, dass das Übertragen der Anforderungen von dem Diagnosepulsanforderungsregister in das Diagnosepulsausführungsregister nur dann erfolgt, wenn in dem Diagnosepulsausführungsregister keine Anforderungen gespeichert sind. Andernfalls wird zunächst eine im Diagnosepulsausführungsregister gespeicherte Anforderung, die als nächste auszuführen ist, in einen an dem zugeordneten Treiberausgang auszuführenden Puls umgesetzt.
- 10 Weiterhin ist nützlich, dass vor dem Übertragen der Anforderungen von dem Diagnosepulsanforderungsregister in das Diagnosepulsausführungsregister in dem Diagnosepulsanforderungsregister nicht mehr gespeicherte Anforderungen ebenfalls in dem Diagnosepulsausführungsregister gelöscht werden. Auf diese Weise wird berücksichtigt, ob Bedingungen, die in der Vergangenheit einmal vorgelegen haben, nach wie vor bestehen; nur wenn dies der Fall ist, kommt eine Ausführung der Pulse am entsprechenden Treiberausgang in Frage.
- 20 Die Erfindung ist in vorteilhafter Weise dadurch weitergebildet, dass nach dem Berücksichtigen einer in dem Diagnosepulsausführungsregister gespeicherten Anforderung diese Anforderung in dem Diagnosepulsausführungsregister gelöscht wird. Bei einer nachfolgenden Abfrage des Diagnosepulsausführungsregisters können somit anderen Treiberausgängen zugeordnete Diagnosepulse ausgeführt werden, sofern entsprechende Anforderungen im Diagnosepulsausführungsregister gespeichert sind.
- 30 Es ist bevorzugt, dass die vorgegebenen Regeln auf mindestens einem der folgenden Kriterien beruhen: verschiedene Treiberausgänge haben unterschiedliche Priorität, und eine einem bestimmten Treiberausgang zugeordnete Anforderung darf berücksichtigt werden oder nicht. Im Falle der Berücksichtigung unterschiedlicher Prioritäten werden die Anforderungen zuerst berücksichtigt, die Treiberausgängen höherer Priorität zugeordnet sind. Beispielsweise kann vorgesehen sein, dass die
- 35

Treiberkanäle in der Reihenfolge ihrer Ordnungszahl priorisiert sind. Im Falle des anderen Kriteriums, dass eine einem bestimmten Treiberausgang zugeordnete Anforderung berücksichtigt werden darf oder nicht, kann vorgesehen sein, in Abhängigkeit sonstiger Voraussetzungen bestimmte Treiberausgänge vollständig von der Beaufschlagung mit angeforderten Pulsen auszunehmen.

Es ist nützlicherweise vorgesehen, dass die Prioritäten der Treiberausgänge durch Konfiguration einer Steuerungs- und Priorisierungseinheit festgelegt werden. Diese kann beispielsweise beeinflussen, ob Stellen in den Registern tatsächlich beschrieben oder gelöscht werden sollen.

Insbesondere ist von Vorteil, dass sich die Prioritäten der Treiberausgänge dynamisch in Abhängigkeit von Betriebszuständen des Kraftfahrzeugs ändern. Beispielsweise kann bei hohen Drehzahlen die Auswertung bestimmter Treiberausgänge besonders sinnvoll sein, die andererseits bei niedrigen Drehzahlen nur von untergeordnetem Interesse sind.

Die Erfindung baut auf dem gattungsgemäßen Diagnosepulsmanager dadurch auf, dass ein Diagnosepulsanforderungsregister zum Speichern einer für einen Treiberausgang bestehenden Anforderung eines Pulses vorgesehen ist, wobei insbesondere mehrere gleichzeitig bestehende Anforderungen gespeichert werden können, und dass mehrere gespeicherte Anforderungen gemäß vorgegebener Regeln nacheinander berücksichtigt werden können. Auf diese Weise werden die Vorteile und Besonderheiten des erfindungsgemäßen Verfahrens auch im Rahmen eines Diagnosepulsmanagers umgesetzt. Dies gilt auch für die nachfolgend angegebenen besonders bevorzugten Ausführungsformen des erfindungsgemäßen Diagnosepulsanforderungsmanagers.

Dieser ist in nützlicher Weise dadurch weitergebildet, dass die Anforderungen in dem Diagnosepulsanforderungsregister als binäre Werte gespeichert werden können, dass die in dem Diag-

nosepulsanforderungsregister gespeicherten Anforderungen in ein Diagnosepulsausführungsregister übertragen werden können und dass das Diagnosepulsanforderungsregister nach dem Übertragen der Anforderungen in das Diagnosepulsausführungsregister gelöscht wird.

Weiterhin ist vorgesehen, dass das Übertragen der Anforderungen von dem Diagnosepulsanforderungsregister in das Diagnosepulsausführungsregister nur dann erfolgt, wenn in dem Diagnosepulsausführungsregister keine Anforderungen gespeichert sind.

Insbesondere ist von Vorteil, dass vor dem Übertragen der Anforderungen von dem Diagnosepulsanforderungsregister in das Diagnosepulsausführungsregister in dem Diagnosepulsanforderungsregister nicht mehr gespeicherte Anforderungen ebenfalls in dem Diagnosepulsausführungsregister gelöscht werden.

Vorzugsweise ist auch vorgesehen, dass nach dem Berücksichtigen einer in dem Diagnosepulsausführungsregister gespeicherten Anforderung diese Anforderung in dem Diagnosepulsausführungsregister gelöscht wird.

Bei dem Diagnosepulsmanager ist es besonders bevorzugt, dass die vorgegebenen Regeln auf mindestens einem der folgenden Kriterien beruhen: verschiedene Treiberausgänge haben unterschiedliche Priorität, und eine einem bestimmten Treiberausgang zugeordnete Anforderung darf berücksichtigt werden oder nicht.

Hierbei ist weiterhin vorgesehen, dass die Prioritäten der Treiberausgänge durch Konfiguration einer Steuerungs- und Priorisierungseinheit festgelegt werden.

Der erfindungsgemäße Diagnosepulsmanager ist in vorteilhafter Weise dadurch weitergebildet, dass sich die Prioritäten der

Treiberausgänge dynamisch in Abhängigkeit von Betriebszuständen des Kraftfahrzeugs ändern.

Der Erfindung liegt die Erkenntnis zugrunde, dass bei gleichzeitiger Anforderung von Pulsen für mehrere Treiberausgänge gewährleistet werden kann, dass alle Pulsanforderungen in einem Diagnosepulsmanager verarbeitet werden. Insbesondere können dabei Priorisierungsregeln und Steuerungsalgorithmen berücksichtigt werden.

10

Die Erfindung wird nun mit Bezug auf die begleitenden Zeichnungen anhand bevorzugter Ausführungsformen beispielhaft erläutert.

15 Dabei zeigt:

Figur 1 ein Funktionsblockdiagramm zur Erläuterung der vorliegenden Erfindung;

20 Figur 2 ein Flussdiagramm zur Erläuterung eines erfindungsgemäßen Verfahrens; und

Figur 3 ein Diagramm zur beispielhaften Erläuterung des Flussdiagramms gemäß Figur 2.

25

Figur 1 zeigt ein Funktionsblockdiagramm zur Erläuterung der vorliegenden Erfindung. Das in Figur 1 dargestellte Zentralelement ist ein Diagnosepulsmanager 10. Dieser ist auf der einen Seite mit Systemanforderungen 12 konfrontiert und muss auf der anderen Seite für eine Pulsausführung 22 sorgen, die sich auf verschiedene Treiberausgänge 20 bezieht. Der Diagnosepulsmanager 10 bewerkstelligt dies durch eine Anforderungsverarbeitung, die insbesondere ein Diagnosepulsanforderungsregister 14 umfasst, und eine Ausführungsverarbeitung, die insbesondere ein Diagnosepulsausführungsregister 16 umfasst. Zwischen der Anforderungsverarbeitung und der Ausführungsverarbeitung wird durch Steuerungs- und Priorisierungsfunktionen

35

18 vermittelt, so dass gewährleistet ist, dass den Systemanforderungen 12 bei der Pulsausführung 22 Rechnung getragen wird, insbesondere unter Berücksichtigung verschiedener Kriterien, was nachfolgend anhand verschiedener Beispiele erläutert wird.

Figur 2 zeigt ein Flussdiagramm zur Erläuterung eines erfindungsgemäßen Verfahrens. Es wird zunächst allgemein ein Ablauf des erfindungsgemäßen Verfahrens erläutert, der nach dem
10 Start des Verfahrens in Schritt S1 mit fester Wiederholrate stattfindet. In Schritt S2 wird geprüft, ob aktuell Diagnosepulse angefordert werden. Ist dies nicht der Fall, so wird in Schritt S2A geprüft, ob in der Vergangenheit Diagnosepulse angefordert wurden. Ist auch dies nicht der Fall, so kehrt
15 der Verfahrensablauf zur Prüfung gemäß Schritt S2 zurück. Wurden in der Vergangenheit jedoch Diagnosepulse angefordert, so geht der Verfahrensablauf von Schritt SA2 zu dem Verfahrensschritt S4 über, der in dem Fall, dass aktuell Diagnosepulse angefordert werden (Schritt S2), über den Verfahrensschritt S3 erreicht wird. In diesem Verfahrensschritt S3 werden die aktuell angeforderten Diagnosepulse in das Diagnosepulsanforderungsregister "diag_puls_anforder_reg" gespeichert.

25 In Schritt S4 wird nun geprüft, ob die Bedingungen für in der Vergangenheit angeforderte Diagnosepulse weiterhin erfüllt sind. Ist dies nicht der Fall, so wird gemäß Schritt S5 im Diagnosepulsausführungsregister "diag_puls_ausführ_reg" und im Diagnosepulsanforderungsregister die betreffende Pulsanforderung gelöscht. Im Anschluss daran beziehungsweise wenn die Bedingungen für in der Vergangenheit angeforderte Diagnosepulse jedoch weiterhin erfüllt sind, wird in Schritt S6 geprüft, ob alle in der Vergangenheit angeforderten Diagnosepulse bereits ausgeführt sind, das heißt, ob für das Diagnosepulsausführungsregister gilt: diag_puls_ausführ_reg == 0.
35 Ist dies der Fall, so werden in Schritt S7 neu angeforderte Diagnosepulse aus dem Diagnosepulsanforderungsregister in das

Diagnosepulsausführungsregister übertragen, das heißt das Diagnosepulsanforderungsregister wird in das Diagnosepulsausführungsregister kopiert ($\text{diag_puls_ausführ_reg} = \text{diag_puls_anforder_reg}$). Nachfolgend wird in Schritt S8 das Diagnosepulsanforderungsregister gelöscht ($\text{diag_puls_anforder_reg} = 0$).

Nachfolgend wird in Schritt S9 der nächste Diagnosepuls aktiviert und im Diagnosepulsausführungsregister wird die betreffende Pulsanforderung gelöscht. Schritt S9 kann auch ausgehend von Schritt S6 direkt erreicht werden, wenn nicht alle in der Vergangenheit angeforderten Diagnosepulse bereits ausgeführt sind. In diesem Fall findet die Aktivierung der nächsten Diagnosepulse und das Löschen der entsprechenden Stelle im Diagnosepulsausführungsregister für in der Vergangenheit angeforderte Diagnosepulse statt. Nach Schritt S9 geht der Funktionsablauf wieder zu Schritt S2 über, beziehungsweise er endet in Schritt S10.

Figur 3 zeigt ein Diagramm zur beispielhaften Erläuterung des Flussdiagramms gemäß Figur 2. Anhand dieses Diagramms werden verschiedene Abläufe bei unterschiedlichen Szenarien von Systemanforderungen erläutert. In Schritt S2 werden zum Zeitpunkt n neue Diagnosepulse angefordert, nämlich für die Treiberausgänge 1, 2 und 8. Folglich wird die in Schritt S2 gemäß Figur 2 gestellte Frage mit "ja" beantwortet, und der Verfahrensablauf geht zu Schritt S3 über.

In Schritt S3 werden unter Vermittlung der Priorisierungs- und Steuereinheit die aktuell angeforderten Pulse im Diagnosepulsanforderungsregister gespeichert. Im beispielhaften Fall werden somit die Registerstellen 1, 2 und 8 auf den Wert 1 gesetzt, wobei das aktive Setzen der Registerstellen im vorliegenden Diagramm gemäß Figur 8 durch Schattierung der Registerstellen gekennzeichnet ist.

Nachfolgend wird in Schritt S4 gefragt, ob die Bedingungen für gegebenenfalls in der Vergangenheit angeforderte Diagnosepulse weiterhin erfüllt sind. Da es im vorliegenden Beispiel keine in der Vergangenheit angeforderten Diagnosepulse gibt, ist die Frage mit "ja" zu beantworten, so dass das Verfahren ohne Aktion in Schritt S5 zu Schritt S6 übergeht. In Schritt S6 wird geprüft, ob gilt: `diag_puls_ausführ_reg == 0`. Dies bedeutet in dem Fall, dass bereits in der Vergangenheit Diagnosepulse angefordert wurden, dass diese auch bereits ausgeführt wurden. Im vorliegenden Beispiel befindet sich das Diagnosepulsausführungsregister durchgängig im Zustand 0, da dies sein Anfangszustand ist. Auf die Frage gemäß Schritt S6 wird somit mit "ja" geantwortet, so dass in Schritt S7 die neu angeforderten Diagnosepulse aus dem Diagnosepulsanforderungsregister in das Diagnosepulsausführungsregister übertragen werden können. Dies erfolgt unter Vermittlung der Priorisierungs- und Steuereinheit.

Nachfolgend wird in Schritt S8 das Diagnosepulsanforderungsregister gelöscht.

In Schritt S9 wird dann der nächste Diagnosepuls aktiviert, und die entsprechende Pulsanforderung wird im Diagnosepulsausführungsregister gelöscht. Im vorliegenden Beispiel hat die Stelle im Diagnosepulsausführungsregister die höchste Priorität, die dem Treiberausgang mit der höchsten Ordnungszahl zugeordnet ist.

Der Verfahrensablauf kehrt nachfolgend zu Schritt S2 zurück, um auf die zum Zeitpunkt $n+1$ vorliegenden Systemanforderungen zu reagieren. Im vorliegenden Beispiel werden wiederum neue Diagnosepulse angefordert, nämlich auch hier wieder für die Treiberausgänge 1, 2 und 8. Die Frage in Schritt S2 wird somit mit "ja" beantwortet, und der Verfahrensablauf geht zu Schritt S3 über. Da der Wert für den Treiberausgang 8 im Diagnosepulsausführungsregister zum Zeitpunkt n in Schritt S9 auf 0 gesetzt wurde, wird dieser in Schritt $n+1$ wieder auf 1

gesetzt, im Gegensatz zu den im Diagnosepulsanforderungsregister den Treiberausgängen 1 und 2 entsprechenden Stellen, da diese im Diagnosepulsausführungsregister nach wie vor auf 1 gesetzt sind.

5

In Schritt S4 wird nachfolgend geprüft, ob die Bedingungen für in der Vergangenheit angeforderte Diagnosepulse weiterhin erfüllt sind. Da dies der Fall ist - für die Treiberausgänge 1, 2 und 8 werden nach wie vor Diagnosepulse angefordert - wird sogleich zu Schritt S6 übergegangen, in dem geprüft wird, ob alle in der Vergangenheit angeforderten Diagnosepulse ausgeführt wurden. Diese Frage wird zum Zeitpunkt $n+1$ mit "nein" beantwortet, da die für den ersten Treiberausgang und den zweiten Treiberausgang angeforderten Diagnosepulse noch nicht ausgeführt wurden. Das Verfahren geht folglich zu Schritt S9 über, wo der in Prioritätsreihenfolge nächste Diagnosepuls ausgeführt wird, nämlich der Diagnosepuls für den zweiten Treiberausgang.

Das Verfahren geht nachfolgend wieder zu Schritt S2 über, wo geprüft wird, ob aktuell Diagnosepulse angefordert werden. Diese Frage wird mit "ja" beantwortet, da zum Zeitpunkt $n+2$ für den Treiberausgang 2 ein Diagnosepuls angefordert wird. Folglich wird in Schritt S3 der aktuell angeforderte Diagnosepuls an der entsprechenden Stelle im Diagnosepulsanforderungsregister gespeichert.

Nachfolgend wird in Schritt S4 geprüft, ob Bedingungen für in der Vergangenheit angeforderte Diagnosepulse weiterhin erfüllt sind. Dies ist nur teilweise der Fall, nämlich für den Treiberausgang 2 und nicht für die Treiberausgänge 1 und 8, so dass die Frage insgesamt mit "nein" beantwortet werden muss. Folglich geht der Verfahrensablauf zu Schritt S5 über, wo im Diagnosepulsausführungsregister und im Diagnosepulsanforderungsregister die entsprechenden Stellen gelöscht werden. Dies betrifft im Diagnosepulsanforderungsregister die Stellen 1 und 8, während im Diagnosepulsausführungsregister

aufgrund der bereits stattgefundenen Impulsausführung am achten Treiberausgang nur die dem ersten Treiberausgang entsprechende Stelle auf 0 gesetzt werden muss.

- 5 In Schritt S6 wird geprüft, ob alle in der Vergangenheit angeforderten Diagnosepulse ausgeführt wurden. Dies ist der Fall, da das Diagnosepulsausführungsregister vollständig auf 0 gesetzt ist. Das Verfahren geht folglich zu Schritt S7 über und der Zustand des Diagnosepulsanforderungsregisters wird in
10 das Diagnosepulsausführungsregister übertragen.

Nachfolgend wird in Schritt S8 das Diagnosepulsanforderungsregister gelöscht.

- 15 In Schritt S9 wird der einzige gemäß Diagnosepulsausführungsregister auszuführende Puls für den zweiten Treiberausgang ausgeführt, und die entsprechende Stelle im Diagnosepulsausführungsregister wird gelöscht.
- 20 Ohne dass im Vorangehenden stets darauf hingewiesen wurde, finden die Veränderungen in den Registern häufig unter Vermittlung der Priorisierung- und Steuerungseinheit statt, was im Diagramm gemäß Figur 3 dadurch veranschaulicht ist, dass die die Beeinflussung der Registerstellen repräsentierenden
25 Pfeile durch die die Priorisierung- und Steuerungseinheit 18 symbolisierenden mit unterbrochenen Linien gezeichneten Kästen hindurchtreten müssen.

- Die Erfindung lässt sich wie folgt zusammenfassen: Es werden
30 ein Verfahren zur Diagnose von Treiberausgängen 20 und ein Diagnosepulsmanager 10 angegeben, die in der Lage sind, einem Treiberausgang 20 einen Puls zuzuführen, um eine Diagnose dieses Treiberausgangs durchzuführen. Da die Systemanforderungen 12 so ausfallen können, dass gleichzeitig mehrere Di-
35 agnosepulse angefordert werden, werden ein Diagnosepulsanforderungsregister 14 und ein Diagnosepulsausführungsregister 16 zur Verfügung gestellt, so dass mehrere gleichzeitig beste-

12

hende Anforderungen 12 gespeichert werden können. Auf der Grundlage dieser Pufferung von Anforderungen 12 ist es möglich, dass mehrere gespeicherte Pulsanforderungen gemäß vorgegebener Regeln nacheinander berücksichtigt werden können.

Patentansprüche

1. Verfahren zur Diagnose von Treiberausgängen (20), insbesondere zum Einsatz in einem Kraftfahrzeug, bei dem zum Aus-
5 lesen eines Diagnoseergebnisses an einem Treiberausgang (20) dem Treiberausgang (20) ein Puls zugeführt wird,
d a d u r c h g e k e n n z e i c h n e t ,
- dass eine für einen Treiberausgang (20) bestehende Anforderung (12) eines Pulses gespeichert werden kann, wobei
10 insbesondere mehrere gleichzeitig bestehende Anforderungen (12) gespeichert werden können, und
 - dass mehrere gespeicherte Anforderungen (12) gemäß vorge-
15 gebener Regeln nacheinander berücksichtigt werden.
2. Verfahren nach Anspruch 1,
d a d u r c h g e k e n n z e i c h n e t ,
- dass die Anforderungen (12) in einem Diagnosepulsanforderungsregister (14) als binäre Werte gespeichert werden
20 können,
 - dass die in dem Diagnosepulsanforderungsregister (14) gespeicherten Anforderungen (12) in ein Diagnosepulsausführungsregister (16) übertragen werden können und
25
 - dass das Diagnosepulsanforderungsregister (14) nach dem Übertragen der Anforderungen (12) in das Diagnosepulsausführungsregister (16) gelöscht wird.
30
3. Verfahren nach Anspruch 2,
d a d u r c h g e k e n n z e i c h n e t ,
dass das Übertragen der Anforderungen (12) von dem Diagnosepulsanforderungsregister (14) in das Diagnosepulsausführungsregister (16) nur dann erfolgt, wenn in dem Diagnosepulsaus-
35

führungsregister (16) keine Anforderungen (12) gespeichert sind.

4. Verfahren nach Anspruch 2 oder 3,
5 d a d u r c h g e k e n n z e i c h n e t ,
 dass vor dem Übertragen der Anforderungen (12) von dem Diagnosepulsanforderungsregister (14) in das Diagnosepulsausführungsregister (16) in dem Diagnosepulsanforderungsregister (14) nicht mehr gespeicherte Anforderungen (12) ebenfalls in
10 dem Diagnosepulsausführungsregister (16) gelöscht werden.
5. Verfahren nach einem der Ansprüche 2 bis 4,
 d a d u r c h g e k e n n z e i c h n e t ,
 dass nach dem Berücksichtigen einer in dem Diagnosepulsausführungsregister (16) gespeicherten Anforderung (12) diese
15 Anforderung (12) in dem Diagnosepulsausführungsregister (16) gelöscht wird.
6. Verfahren nach einem der vorhergehenden Ansprüche,
20 d a d u r c h g e k e n n z e i c h n e t ,
 dass die vorgegebenen Regeln auf mindestens einem der folgenden Kriterien beruhen:
 - verschiedene Treiberausgänge (20) haben unterschiedliche
25 Priorität, und
 - eine einem bestimmten Treiberausgang (20) zugeordnete Anforderung (12) darf berücksichtigt werden oder nicht.
- 30 7. Verfahren nach Anspruch 6,
 d a d u r c h g e k e n n z e i c h n e t ,
 dass die Prioritäten der Treiberausgänge (20) durch Konfiguration einer Steuerungs- und Priorisierungseinheit (18) festgelegt werden.
- 35 8. Verfahren nach Anspruch 6 oder 7,
 d a d u r c h g e k e n n z e i c h n e t ,

dass sich die Prioritäten der Treiberausgänge (20) dynamisch in Abhängigkeit von Betriebszuständen des Kraftfahrzeugs ändern.

- 5 9. Diagnosepulsmanager (10) zum Zuführen von Pulsen zu Treiberausgängen (20) in Abhängigkeit von Systemanforderungen (12), insbesondere zum Einsatz in einem Kraftfahrzeug, um an Treiberausgängen (20) Diagnoseergebnisse auszulesen, d a d u r c h g e k e n n z e i c h n e t ,
- 10
- dass ein Diagnosepulsanforderungsregister (14) zum Speichern einer für einen Treiberausgang (20) bestehenden Anforderung (12) eines Pulses vorgesehen ist, wobei insbesondere mehrere gleichzeitig bestehende Anforderungen
 - 15 (12) gespeichert werden können, und
 - dass mehrere gespeicherte Anforderungen (12) gemäß vorgegebener Regeln nacheinander berücksichtigt werden können.
- 20 10. Diagnosepulsmanager (10) nach Anspruch 9, d a d u r c h g e k e n n z e i c h n e t ,
- dass die Anforderungen (12) in dem Diagnosepulsanforderungsregister (14) als binäre Werte gespeichert werden
 - 25 können,
 - dass die in dem Diagnosepulsanforderungsregister (14) gespeicherten Anforderungen (12) in ein Diagnosepulsausführungsregister (16) übertragen werden können und
 - 30
 - dass das Diagnosepulsanforderungsregister (14) nach dem Übertragen der Anforderungen (12) in das Diagnosepulsausführungsregister (16) gelöscht wird.
- 35 11. Diagnosepulsmanager (10) nach Anspruch 10, d a d u r c h g e k e n n z e i c h n e t ,

16

- dass das Übertragen der Anforderungen (12) von dem Diagnosepulsanforderungsregister (14) in das Diagnosepulsausführungsregister (16) nur dann erfolgt, wenn in dem Diagnosepulsausführungsregister (16) keine Anforderungen (12) gespeichert sind.

12. Diagnosepulsmanager (10) nach Anspruch 10 oder 11,
dadurch gekennzeichnet,
dass vor dem Übertragen der Anforderungen (12) von dem Diagnosepulsanforderungsregister (14) in das Diagnosepulsausführungsregister (16) in dem Diagnosepulsanforderungsregister (14) nicht mehr gespeicherte Anforderungen (12) ebenfalls in dem Diagnosepulsausführungsregister (16) gelöscht werden.
13. Diagnosepulsmanager (10) nach einem der Ansprüche 10 bis 12,
dadurch gekennzeichnet,
dass nach dem Berücksichtigen einer in dem Diagnosepulsausführungsregister (16) gespeicherten Anforderung (12) diese Anforderung (12) in dem Diagnosepulsausführungsregister (16) gelöscht wird.
14. Diagnosepulsmanager (10) nach einem der Ansprüche 9 bis 13,
dadurch gekennzeichnet,
dass die vorgegebenen Regeln auf mindestens einem der folgenden Kriterien beruhen:
- verschiedene Treiberausgänge (20) haben unterschiedliche Priorität und
 - eine einem bestimmten Treiberausgang (20) zugeordnete Anforderung (12) darf berücksichtigt werden oder nicht.
15. Diagnosepulsmanager (10) nach Anspruch 14,
dadurch gekennzeichnet,

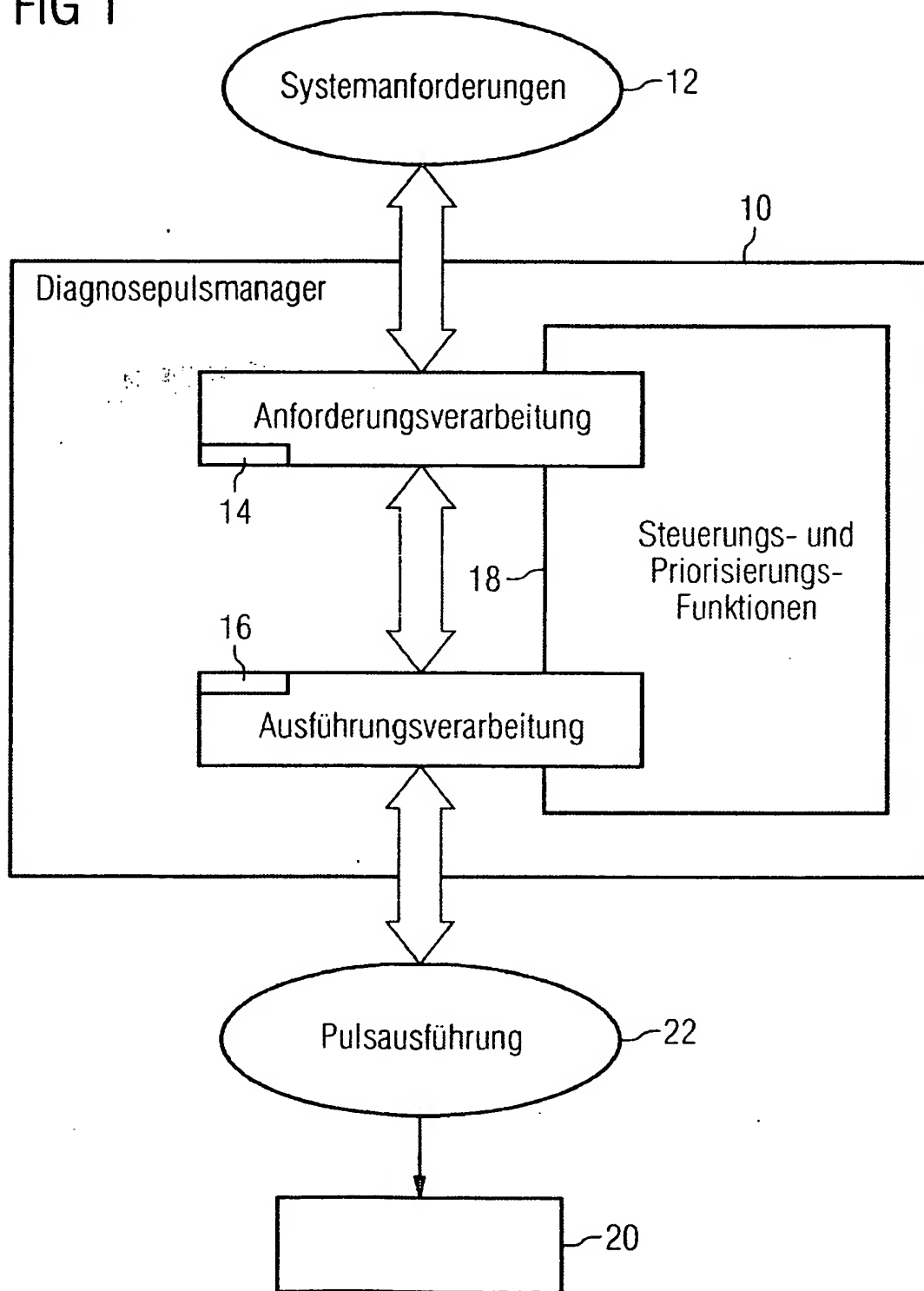
dass die Prioritäten der Treiberausgänge (20) durch Konfiguration einer Steuerungs- und Priorisierungseinheit (18) festgelegt werden.

- 5 16. Diagnosepulsmanager (10) nach Anspruch 14 oder 15,
d a d u r c h g e k e n n z e i c h n e t ,
dass sich die Prioritäten der Treiberausgänge (20) dynamisch in Abhängigkeit von Betriebszuständen des Kraftfahrzeugs ändern.

THIS PAGE BLANK (USPTO)

1/6

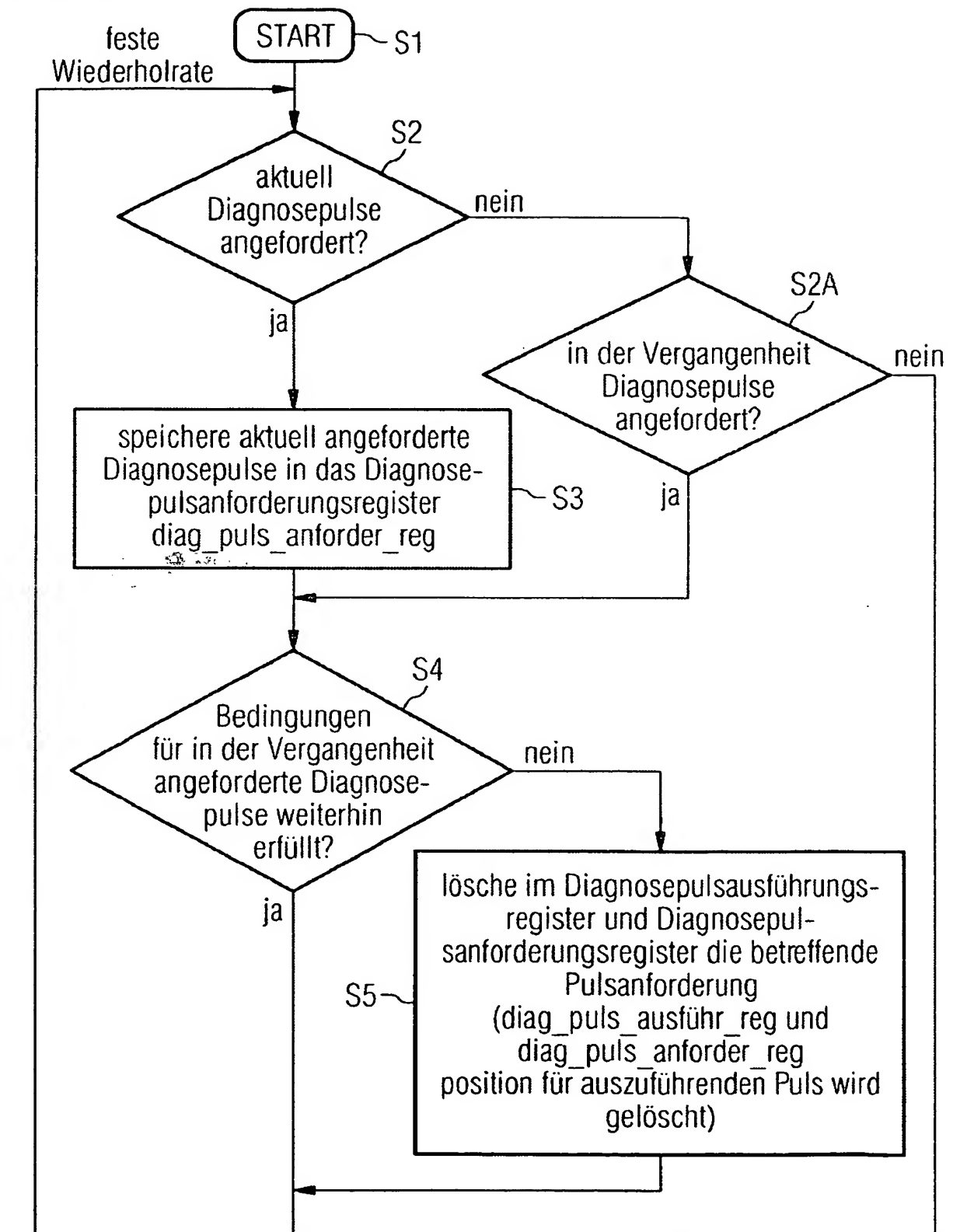
FIG 1



THIS PAGE BLANK (USE)

2/6

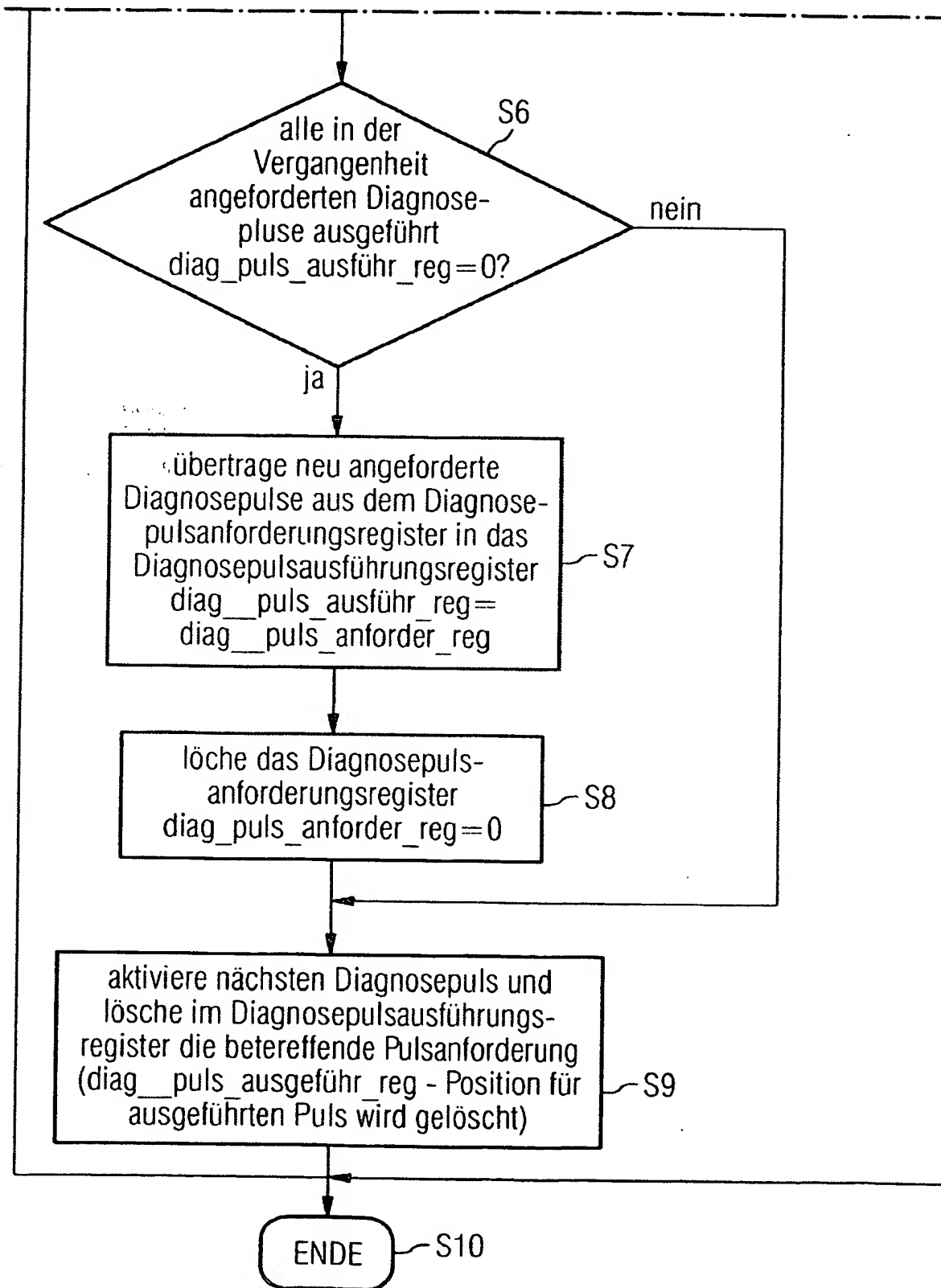
FIG 2A



THIS PAGE BLANK (US)

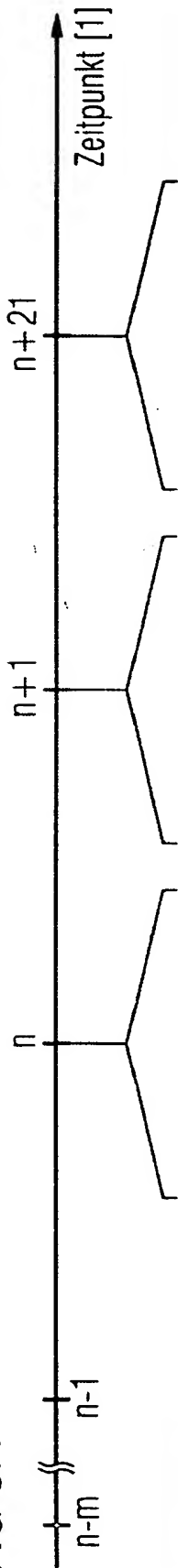
3/6

FIG 2B



THIS PAGE BLANK (US)

FIG 3A

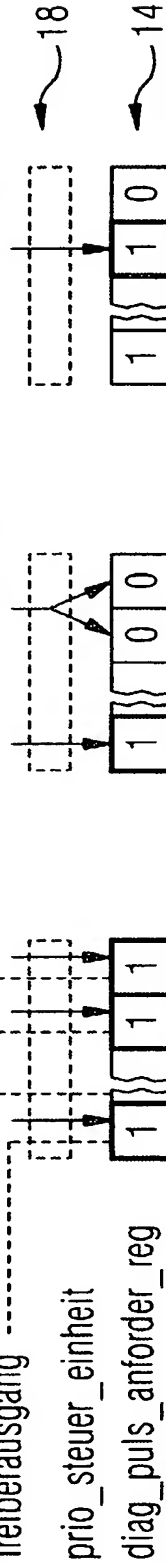


S2

neue Diagnosepulse angefordert -> ja für Treiberausgang 1,2 und 8	neue Diagnosepulse angefordert -> ja für Treiberausgang 1,2 und 8	neue Diagnosepulse angefordert -> ja für Treiberausgang 2
---	---	---

1. Treiberausgang
2. Treiberausgang
8. Treiberausgang
prio_steuer_einheit
diag_puls_anforder_reg

für Treiberausgang 1 und 2 wird d. diag_puls_anforder_reg nicht beschrieben, da die Pulsanforderung zum Zeitpunkt n für Treiberausgang 1 und 2 noch nicht ausgeführt wurde



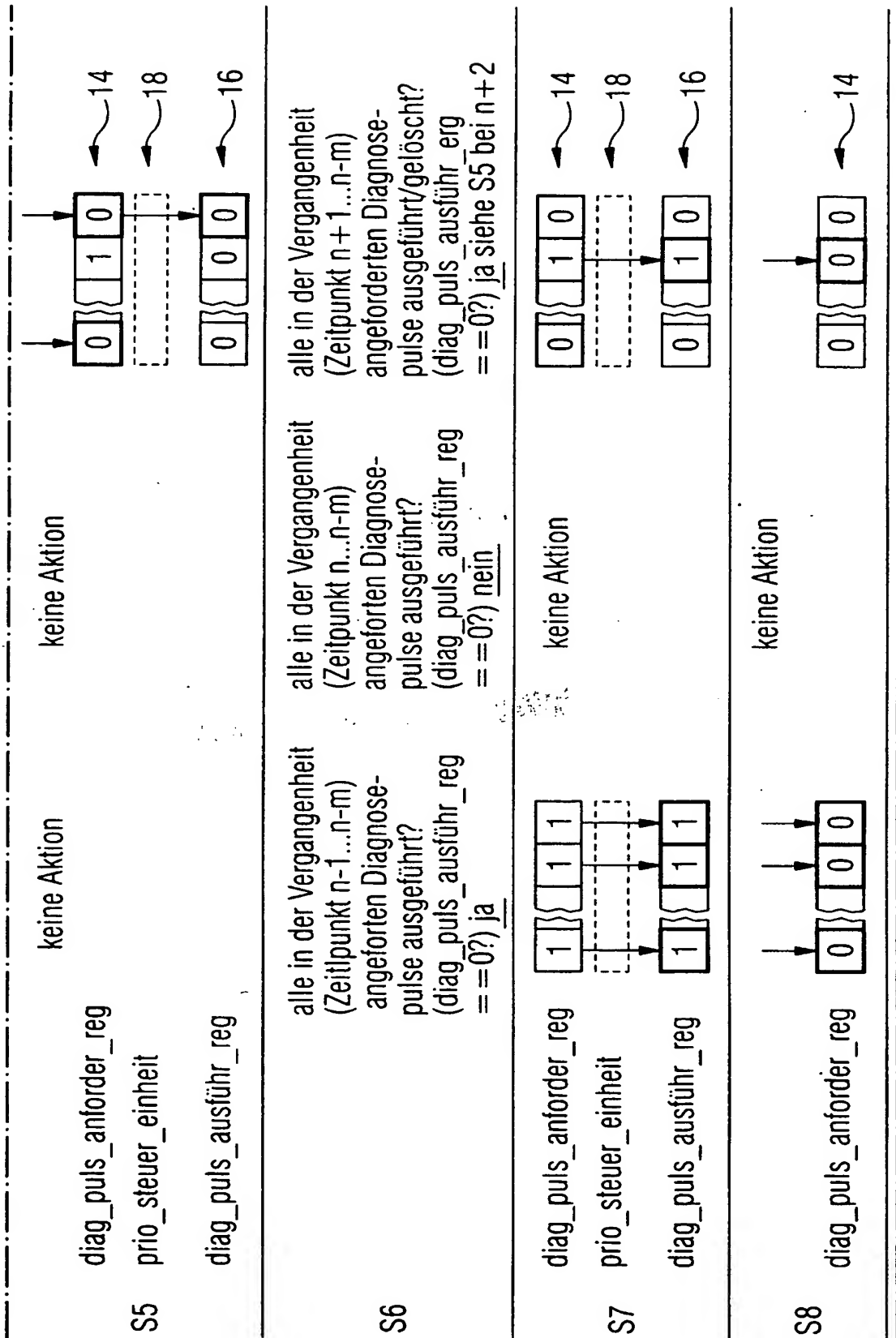
S3

keine Aktion	keine Aktion	Beispiel: Bedingungen für in der Vergangenheit (Zeitpunkt $n+1 \dots n-m$) angeforderte Diagnosepulse weiterhin erfüllt? für Treiberausgang 1 und 8 <u>nein</u> für Treiberausgang 2 ja
--------------	--------------	---

S4

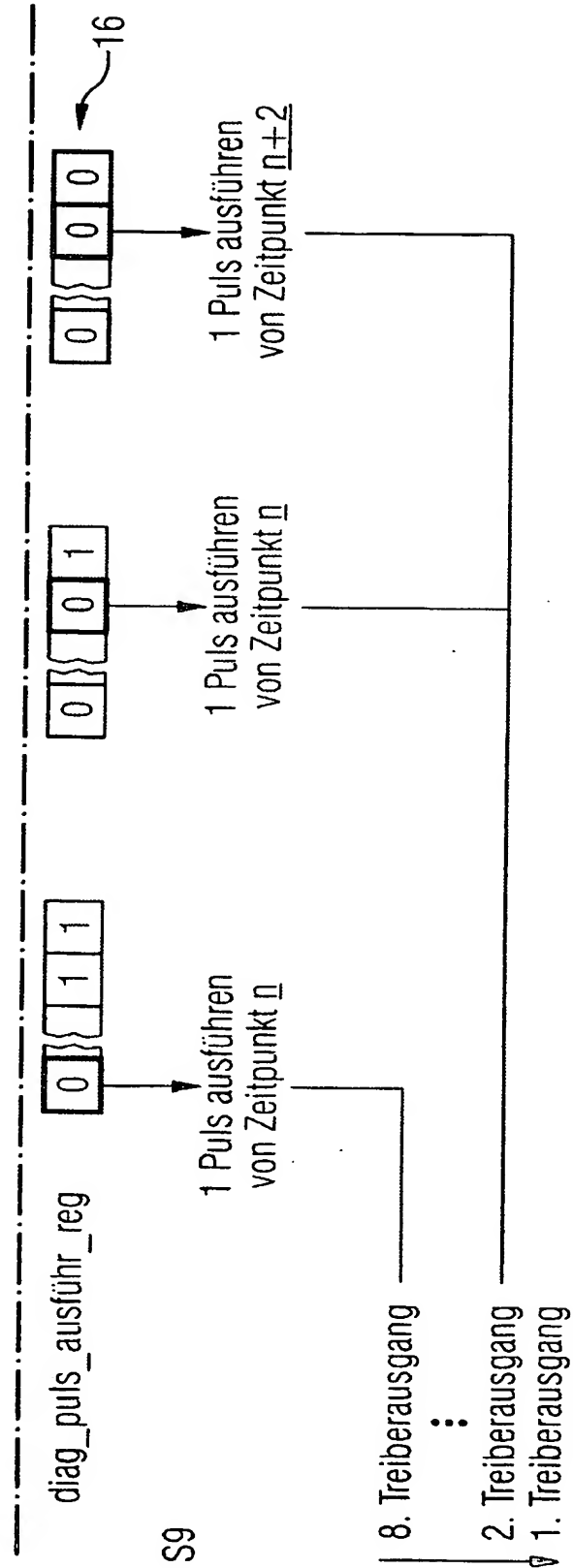
THIS PAGE BLANK (USPTO)

FIG 3B



THIS PAGE BLANK (USPTO)

FIG 3C



THIS PAGE BLANK (USPTO)

Docket # S4-D3 P07726

Applic. # _____

Applicant: Läufel, Andreus et al

Lerner Greenberg Sterner LLP
Post Office Box 2480

Hollywood, FL 33022-2480

Tel: (954) 925-1100 Fax: (954) 925-1101